AIDA Framework: Real-Time Correlation and Prediction of Intrusion Detection Alerts

CyberTIM 2019

Wednesday 28th August, 2019

Martin Husák
Jaroslav Kašpar
Introduction – Information Sharing

Information Sharing in Cyber Security
- Collaboration and information exchange are fundamental to cyber security
- Automated, effective, and efficient information sharing is still problematic
- Information sharing platforms, e.g., SABU (https://sabu.cesnet.cz)

Analysis of Security Alerts
- Large volumes of data from IDS, honeypots, blacklists, …
- Heterogeneity of the data – alerts, IoC, vulnerabilities, …
- Unclear goals – what to do with the data?
Introduction – SABU Platform

Intrusion Detection System

Honeypot

Third-party Alert Sharing Platform

Active Network Defense (firewalls, RTBH, ...)

Research & Threat Intelligence

AIDA Framework
Page 3 / 19
Motivation – Blacklisting and Predictions

**Personalized Blacklisting**
- Receivers of the data are typically interested only in small fraction of them.
- Receivers are not capable of responding to every information in the sharing platform.
- Weekly reports are personalized, but the data are from the past.
- Receivers need small number of items (e.g., IP addresses) that they can react to.

**Predictions and projections**
- Predicting that an attack will occur, e.g., by time series analysis.
- Projecting the next step of an attacker, e.g., attack matching a known pattern.
- Personalized blacklist can be based on predicted and projected attacks.
AIDA Framework
AIDA Framework

Purpose
- Analytical framework for processing intrusion detection alerts.
- Motivated by the needs and development of the SABU platform.
- Predictive analytics – attack projections based on historical observations.

Design
- Big Data approaches – stream processing.
- Data mining used to infer predictive rules.
- Complex event processing-inspired rule matching (predictions).
AIDA Framework – Schema

Input → kafka → Output

- Alerts
- Data Sanitizer
- Alert Aggregation
- Data Mining
- Sequential Rules
- Rule Matching
- Predicted Alerts

Feedback
AIDA Framework – Data Distribution

Inputs and Outputs

- Deployed version uses Warden client to communicate with the SABU platform.
- Receiving connectors receives alerts, sending connector sends predicted alerts.

Kafka message broker

- Distributes the data in topics to the framework components.
- Ensures correct data order of data processing.
AIDA Framework – Components

Data Sanitization
- Semantic checks – filtering testing messages, alerts with no IP addresses, etc.

Alert Aggregation
- Aggregation of multiple copies of the same alert.
- Aggregation of repeatedly reported events in different time.
AIDA Framework – Information Extraction

Data Mining
- Top-k sequential rule mining.
- Using algorithms implemented in SPMF library.

Predictive rule example

OrganizationA.Honeypot1_Recon.Scanning_22, OrganizationB.IDS1_Attempt.Login_22

=>$OrganizationA.IDS1_Attempt.Login_22

#SUPP: 0.0011 #CONF: 0.6111
AIDA Framework – Prediction

Rule Matching
- Based on Esper – Complex Event Processing engine.
- Esper EPL – SQL-like data stream querying language.
- Predictive rules are converting to EPL queries.
- If an EPL query finds a match, a new alert is predicted.

Feedback
- Simple counter and logger of processed and predicted alerts.
AIDA Framework – Dashboard
AIDA Framework – Dashboard

Rules

Active rules

Add new rule
Deployment
Deployment in SABU

Intrusion Detection System

Honeypot

Third-party Alert Sharing Platform

iABU

Active Network Defense (firewalls, RTBH, ...)

Research & Threat Intelligence

AIDA Framework
Page 15 / 19
Deployment in SABU

Data volume and performance
- 1.7 million alerts per day.
- Commodity hardware – 8 CPUs, 16 GB RAM.
- Up to hundred EPL queries running in parallel.

Sample results
- 1.7 million alerts produces around 650,000 sequences.
- Around 55% of alerts are aggregated.
- Top-10 rules mined every day, approx. 80% are usable
- Rule confidence most frequently around 0.7, often up to 0.9.
Stand-alone deployment

Running AIDA locally

- AIDA Framework is distributed with a Vagrant file.
- Automated deployment in a virtual machine.
- Still, it is needed to manually trigger data mining and load predictive rules.

Use Case

- Experimentations over datasets
- A sample dataset with alerts from SABU platform was published at http://dx.doi.org/10.17632/p6tym3fghz.1
Conclusion
Conclusion

AIDA Framework
- Analytical framework for processing intrusion detection alerts
- Inspired by the needs of SABU alert sharing platform
- Data mining-supported extraction of common attack patterns
- Predictions of attack continuations; personalized blacklisting

Deployment and Usage
- Operational deployment in the SABU platform
- Stand-alone deployment for experimentation
HTTPS://GITHUB.COM/CSIRT-MU/AIDA-FRAMEWORK

Martin Husák
husakm@ics.muni.cz

@csirtmu
sabu.cesnet.cz